Fast modified discrete cosine transform method

Field of the Invention

This invention is applicable in the field of multi-channel audio coders which use a modified discrete cosine transform as a step in the compression of audio signals.

Background of the Invention

Recent years have witnessed an unprecedented advancement in audio coding technology. This has led to high compression ratios while keeping audible degradation in the compressed signal to a minimum. These coders are intended for a variety of applications, including 5.1 channel film soundtracks, HDTV, laser discs and multimedia. Description of a compression method used in such coders can be found in the ATSC Standard, *"Digital Audio Compression (AC-3) Standard"*, Document A/52, 20 December, 1995.

In the basic approach, at the encoder the time domain signal is sectioned into frames, each frame comprising of a number of blocks. The time domain signal in each block is first converted to the frequency domain using a bank of filters. The frequency domain coefficients, thus generated, are next converted to floating point representation. In floating point syntax, each coefficient is represented as a mantissa and an exponent. The bulk of the compressed bitstream transmitted to the decoder comprises these exponents and mantissas.

The coders are expected to operate in real time. Therefore it is important that the processing at both the encoders and decoders be highly optimised. The ATSC standard outlines a method for conversion of the inverse cosine transform to Fast Fourier Transform to obtain remarkable speed gain.

Frequency transformation presents one of the greatest computation requirements in any transform coder. Therefore, an efficient implementation of this phase can decrease the

computation requirement of the system significantly and make real time operation of encoder an easily attainable possibility.

In some encoders such as the AC-3, the frequency domain transformation of signals is performed by the modified discrete cosine transform (MDCT). If directly implemented, the MDCT requires $O(N^2)$ additions and multiplications.

However it is possible to reduce the number of required operations significantly if one is able to express the MDCT equation into a form that is amenable to the use of the well known Fast Fourier Transform (FFT) method of J.W. Cooley and J.W. Tukey (1960).

To compute the MDCT of a sequence of N real data, conventional methods use a system comprising of a pre-processing block, an N-point FFT block and finally a post-processing block.

It is an object of the invention to further reduce the computation required to evaluate the MDCT.

Summary of the Invention

In accordance with the invention, there is provided a method of evaluating a modified discrete cosine transform (MDCT) of an input sequence of complex data using a Fast Fourier Transform (FFT), wherein the FFT is a N/2 - Point FFT.

Preferably, the MDCT is of the form:

$$X_k = \sum_{n=0}^{n=N-1} x[n] * \cos(2\pi * (2n+1) * (2k+1)/4N + \pi * (2k+1)/4) \quad k = 0...(N/2-1)$$

where $x[n]$ represents the input sequence and N is the number of data samples in the sequence.

Preferably, a transform coefficient sequence of the MDCT is derived using the N/2 - Point FFT of a complex function ($z[n]$, where

$$z[n] = (x[2n] + jx[2n+1])e^{j\frac{2\pi n}{N}}, \quad 0 \le n \le (N/2-1)$$

Preferably, a transform Zk and the complex conjugate of the transform $Z_{N/2-k-1}$ are derived from z[n] for the expression

$$G_k = \frac{1}{2}\left[Z_k + Z^*_{\frac{N}{2}-k-1} \right] + \frac{e^{j\frac{\pi(2k+1)}{N}}}{2j}\left[Z_k - Z^*_{\frac{N}{2}-k-1} \right]$$

where Gk is a transform coefficient sequence which allows the MDCT to be evaluated, in that

$$G_k = g_{k,r} + jg_{k,i} = \sum_{n=0}^{n=N-1} (x[n]e^{j\pi n/N}) * e^{j2\pi nk/N}$$

which can be used to solve the MDCT when rewritten as

$$X_k = \cos\gamma(k)\underbrace{1/2(e^{j\pi(k+1/2)/N}G_k + e^{-j\pi(k+1/2)/N}G^*_k)}_{T_1} - \sin\gamma(k)\underbrace{1/2j(e^{j\pi(k+1/2)/N}G_k - e^{-j\pi(k+1/2)/N}G^*_k)}_{T_2}$$

$$= \cos\gamma^*\underbrace{(g_{k,r}\cos(\pi(k+1/2)/N) - g_{k,i}\sin(\pi(k+1/2)/N)}_{T_1}$$

$$-\sin\gamma^*\underbrace{(g_{k,r}\sin(\pi(k+1/2)/N) + g_{k,i}\cos(\pi(k+1/2)/N)}_{T_2} \qquad \text{Eq. 16}$$

Brief Description of the Drawings

The invention is more fully described, by way of non-limiting example only, with reference to the drawings, in which:

Figure 1 is a representation of a an audio input sequence;

Figure 2 is a diagrammatic representation of an encoder; and

Figure 3 illustrates evaluation of the input sequence, into a coefficient set.

Detailed Description of a Preferred Embodiment

The input to an audio coder comprises a stream of digitised samples of a time domain analog signal. For a multi-channel encoder the stream consists of interleaved samples for each channel. The input stream is sectioned into blocks, each block containing N consecutive

samples of each channel (see Fig. 1). Thus within a block the N samples of a channel form a sequence $\{x[0],x[1],x[2],\ldots,x[N-1]\}$. The time domain samples are next converted to the frequency domain using an analysis filter bank (see Fig. 2).

The frequency domain coefficients, thus generated, form a coefficient set which can be identified as $(X_0,X_1,X_2,\ldots,X_{N/2-1})$. Since the signal is real only the first $N/2$ frequency components are considered. Here X_0 is the lowest frequency (DC) component while $X_{N/2-1}$ is the highest frequency component of the signal.

Audio compression essentially entails finding how much of the information in the set $(X_0,X_1,X_2,\ldots,X_{N/2-1})$ is necessary to reproduce the original analog signal at the decoder with minimal audible distortion.

The coefficient set is normally converted into floating point format, where each coefficient is represented by an exponent and mantissa .

The exponent set is usually transmitted in its original form. However, the mantissa is truncated to a fixed or variable number of decimal places. The value of number of bits for coding a mantissa is usually obtained from a *bit allocation algorithm* which for advanced psychoacoustic coders may be based on the masking property of the human auditory system. A low number of bits results in higher compression ratio because less space is required to transmit the coefficients. However this causes very high quantization error leading to audible distortion. A good distribution of available bits to each mantissa forms the core of the most advanced encoders.

In some encoders such as the AC-3, the frequency domain transformation of signals is performed by the (MDCT) modified discrete cosine transform (Eq. 1).

$$X_k = \sum_{n=0}^{n=N-1} x[n] * \cos(2\pi * (2n+1) * (2k-1)/4N - \pi * (2k+1)/4) \quad k=0\ldots(N/2-1) \qquad \text{Eq. 1}$$

If directly implemented in the form given above, the MDCT requires $O(N^2)$ additions and multiplications.

However it is possible to reduce the number of required operations significantly if one is able to evaluate Eq. 1 using the well known Fast Fourier Transform method of J.W. Cooley and J.W. Tukey (1960). The general Discrete Fourier Transform (DFT) is given below (Eq. 2). It requires $O(N^2)$ complex additions and multiplications. But by using the Fast Fourier Transform method the DFT in Eq. 2 can be computed with $O(N\log_2 N)$ operations only.

Eq.2

$$F_k = \sum_{n=0}^{n=N-1} (x[n] * e^{2\pi jnk/N}) \quad k=0...N-1 \qquad \text{Eq.2}$$

Here j is the symbol for imaginary number, i.e. $j=\sqrt{-1}$.

Although it may not be immediately apparent how Eq. 1 can be transformed to Eq. 2, a careful analysis shows that this is indeed possible. To simplify Eq. 1, let us define two functions

$$\alpha(n,k) = 2\pi(2n+1)(2k+1)/4N \qquad \text{Eq. 3}$$

and

$$\Upsilon(k) = \pi(2k+1)/4 \qquad \text{Eq. 4}$$

Then, using these functions, Eq. 1 can be rewritten as

$$X_k = \sum_{n=0}^{n=N-1} x[n] * \cos(\underbrace{2\pi * (2n+1) * (2k+1)/4N}_{\alpha(n,k)} + \underbrace{\pi * (2k+1)/4)}_{\gamma(k)} \quad k = 0...(N/2-1)$$

$$= \sum_{n=0}^{n=N-1} x[n] * \cos(\alpha(n.k) + \gamma(k)) \qquad \text{Eq. 5}$$

$$= \sum_{n=0}^{n=N-1} x[n] * (\cos\alpha(n,k)\cos\gamma(k) - \sin\alpha(n,k)\sin\gamma(k)) \qquad \text{Eq. 6}$$

In Eq.6 the trigonometric equality, $\cos(a+b)=\cos a \cos b - \sin a \sin b$ is used for simplification. Furthermore, since the function $\gamma(k)$ is only dependent on variable n, it can be brought outside the summation expression to give

$$X_k = \cos\gamma(k) \underbrace{\sum_{n=0}^{n=N-1} x[n] * \cos\alpha(n,k)}_{T_1} - \sin\gamma(k) \underbrace{\sum_{n=0}^{n=N-1} x[n] * \sin\alpha(n,k)}_{T_2} \qquad \text{Eq. 7}$$

The two terms, T_1 and T_2, in Eq.7 can now be evaluated separately. From Euler's identity $e^{j\theta} = \cos\theta + j\sin\theta$, we know that $\cos\alpha(n,k) = (e^{j\alpha(n,k)} + e^{-j\alpha(n,k)})/2$ and $\sin\alpha(n,k) = (e^{j\alpha(n,k)} - e^{-j\alpha(n,k)})/2j$. Therefore we can rewrite the terms T_1 and T_2 as

$$T_1 = \sum_{n=0}^{n=N-1} x[n] * (e^{j\alpha} + e^{-j\alpha})/2 = 1/2\{\underbrace{\sum_{n=0}^{n=N-1} x[n] * e^{j\alpha}}_{A_1} + \underbrace{\sum_{n=0}^{n=N-1} x[n] * e^{-j\alpha}}_{A_2}\} \qquad \text{Eq. 8}$$

$$T_2 = \sum_{n=0}^{n=N-1} x[n] * (e^{j\alpha} - e^{-j\alpha})/2j = 1/2j\{\underbrace{\sum_{n=0}^{n=N-1} x[n] * e^{j\alpha}}_{A_1} - \underbrace{\sum_{n=0}^{n=N-1} x[n] * e^{-j\alpha}}_{A_2}\} \qquad \text{Eq. 9}$$

We evaluate the term A_1. From Eq. 8 and Eq. 9 note that

$$\begin{aligned} A_1 &= \sum_{n=0}^{n=N-1} x[n] * e^{j\alpha} \\ &= \sum_{n=0}^{n=N-1} x[n] * e^{j(2\pi(2n+1)(2k+1)/4N} \\ &= e^{j\pi(k+1/2)/N} * \sum_{n=0}^{n=N-1} (x[n] * e^{j\pi n/N}) * e^{j2\pi nk/N} \end{aligned} \qquad \text{Eq. 10}$$

If we define a complex variable

$$x'[n] = x[n] * e^{j\pi n/N} \qquad \text{Eq. 11}$$

then the Eq. 10 is simply

$$A_1 = e^{j\pi(k+1/2)/N} * \underbrace{\sum_{n=0}^{n=N-1} x'[n] * e^{j2\pi nk/N}}_{G_k} \qquad \text{Eq. 12}$$

The complex term $G_k = g_{k,r} + j g_{k,i}$, where $g_{k,r}$ and $g_{k,i} \in \Re$ (set of real numbers) in Eq. 12 is essentially the same as F_k in Eq. 2. Therefore the FFT approach can be used to evaluate G_k. This brings down the computation from O/N^2 to $O(N\log_2 N)$. We can now, similarly, evaluate

the second term A_2 in Eq. 8 and Eq. 9.

$$A_2 = \sum_{n=0}^{n=N-1} x[n] * e^{-j\alpha(n,k)} = e^{-j\pi(k+1/2)/N} * \underbrace{\sum_{n=0}^{n=N-1} (x[n] * e^{-j\pi n/N}) * e^{-j2\pi nk/N}}_{G_k^*} \qquad \text{Eq. 13}$$

Note that G_k^* is actually the complex conjugate of G_k which was obtained earlier by Eq. 12. That is, if $G_k = g_{k,r} + jg_{k,i}$, where $g_{k,r}$ and $g_{k,i} \in \Re$ as defined earlier then $G_k^* = g_{k,r} + jg_{k,i}$. Therefore G_k^* in Eq. 12 does not need to computed again and result from Eq. 12 can be reused. That is only one FFT needs to be computed for the evaluation of T_1. Using Eq. 8 to 13 we obtain

$$T_1 = 1/2(e^{j\pi(k+1/2)/N} G_k + e^{-j\pi(k+1/2)/N} G_k^*) \qquad \text{Eq. 14}$$

Next, we analyse the term T_2.

$$\begin{aligned} T_2 &= \sum_{n=0}^{n=N-1} x[n] * (e^{j\alpha} - e^{-j\alpha})/2j \\ &= 1/2j(A_1 - A_2) \\ &= 1/2j(e^{j\pi(k+1/2)/N} G_k - e^{-j\pi(k+1/2)/N} G_k^*) \end{aligned} \qquad \text{Eq. 15}$$

Finally after simplifications of Eq. 7,14 and 15 we obtain

$$\begin{aligned} X_k &= \cos\gamma(k) \underbrace{1/2(e^{j\pi(k+1/2)/N} G_k + e^{-j\pi(k+1/2)/N} G_k^*)}_{T_1} - \sin\gamma(k) \underbrace{1/2j(e^{j\pi(k+1/2)/N} G_k - e^{-j\pi(k+1/2)/N} G_k^*)}_{T_2} \\ &= \cos\gamma * \underbrace{(g_{k,r}\cos(\pi(k+1/2)/N) - g_{k,i}\sin(\pi(k+1/2)/N)}_{T_1} \\ &\quad - \sin\gamma * \underbrace{(g_{k,r}\sin(\pi(k+1/2)/N) + g_{k,i}\cos(\pi(k+1/2)/N)}_{T_2} \end{aligned} \qquad \text{Eq. 16}$$

The term $G_k = g_{k,r} + jg_{k,i}$ may be computed in O(NlogN) operation by using FT algorithms. The additional operation outlined in Eq. 16 to extract the final X_k is only of order $O(N)$. Therefore the MDCT can now be computed in $O(N\log_2 N)$ time. The operations required to obtain the MDCT is illustrated in Fig. 3.

N-Point MDCT Using N/2-Point FFT

Since the most computationally intensive part is the FFT, we shall focus on it in order to decrease computation even further. The values of $G_k = \sum_{n=0}^{N-1}\left[\left(x[n]e^{j\frac{\pi n}{N}}\right)e^{j\frac{2\pi nk}{N}}\right]$ required in Eq. 16 may be obtained through an N/2-Point FFT.

It is well known that an N-Point FFT of a sequence of N real data can be easily obtained by using a N/2-Point FFT only, provided the FFT is preceded by a pre-processing and followed by a post-processing which formats the inputs so that the final result is mathematically identical in both cases.

In our present case, the input to the FFT block is a sequence of N complex data. To evaluate the FFT of N arbitrary complex numbers, an N-Point FFT is definitely required. A M-Point FFT, M < N will not do. In the present case however, the input sequence is not completely an arbitrary complex sequence but is of the form $x[n]e^{j\frac{\pi n}{N}}$, where x[n] is real.

$$G_k = \sum_{n=0}^{N-1}\left[\left(x[n] * e^{j\frac{\pi n}{N}}\right)e^{j\frac{2\pi nk}{N}}\right]$$

Let us define $z[n] = (x[2n] + jx[2n+1])e^{j\frac{2\pi n}{N}}$, $0 \le n \le (N/2-1)$. Then, taking a N/2-Point FFT of z[n],

$$Z_k = \sum_{n=0}^{N/2-1}\left[z[n]e^{j\frac{2\pi nk}{N/2}}\right], \quad 0 \le k \le N/2-1$$

Therefore, $Z_{\frac{N}{2}-k-1} = \sum_{n=0}^{N/2-1}\left[z[n]e^{-j\frac{2\pi nk}{N/2}}e^{-j\frac{2\pi n}{N/2}}\right]$. Taking the complex conjugate of $Z_{\frac{N}{2}-k-1}$ gives us $Z^*_{\frac{N}{2}-k-1} = \sum_{n=0}^{N/2-1}\left[z^*[n]e^{j\frac{2\pi nk}{N/2}}e^{j\frac{2\pi n}{N/2}}\right]$. Also, note that with some simplifications we obtain $z[n] + z^*[n]e^{j\frac{2\pi n}{N/2}} = 2x[2n]e^{j\frac{\pi(2n)}{N}}$ and $z[n] - z^*[n]e^{j\frac{2\pi n}{N}} = 2x[2n]e^{j\frac{\pi(2n)}{N}}$. We now have all the essential derivations for us to obtain G_k from Z_k.

$$\frac{1}{2}\left[Z_k + Z^*_{\frac{N}{2}-k-1}\right] = \frac{1}{2}\left[\sum_{n=0}^{N/2-1}\left[z[n]e^{j\frac{2\pi nk}{N/2}}\right] + \sum_{n=0}^{N/2-1}\left[z^*[n]e^{j\frac{2\pi nk}{N/2}}e^{j\frac{2\pi n}{N/2}}\right]\right]$$

$$= \frac{1}{2}\left[\sum_{n=0}^{N/2-1}\left[\left(z[n] + z^*[n]e^{j\frac{2\pi n}{N/2}}\right)e^{j\frac{2\pi nk}{N/2}}\right]\right]$$

$$= \frac{1}{2}\sum_{n=0}^{N/2-1}\left[\left(2x[2n]e^{j\frac{\pi(2n)}{N}}\right)e^{j\frac{2\pi(2n)k}{N}}\right]$$

$$= \sum_{n=0}^{N/2-1}\left[\left(x[2n]e^{j\frac{\pi(2n)}{N}}\right)e^{j\frac{2\pi(2n)k}{N}}\right]$$

$$\frac{e^{j\frac{\pi(2k+1)}{N}}}{2j}\left[Z_k - Z^*_{\frac{N}{2}-k-1}\right] = \frac{e^{j\frac{\pi(2k+1)}{N}}}{2j}\left[\sum_{n=0}^{N/2-1}\left[z[n]e^{j\frac{2\pi nk}{N/2}}\right] - \sum_{n=0}^{N/2-1}\left[z^*[n]e^{j\frac{2\pi nk}{N/2}}e^{j\frac{2\pi n}{N/2}}\right]\right]$$

$$= \frac{e^{j\frac{\pi(2k+1)}{N}}}{2j}\left[\sum_{n=0}^{N/2-1}\left[\left(z[n] - z^*[n]e^{j\frac{2\pi n}{N/2}}\right)e^{j\frac{2\pi nk}{N/2}}\right]\right]$$

$$= \frac{e^{j\frac{\pi(2k+1)}{N}}}{2j}\left[\sum_{n=0}^{N/2-1}\left(2jx[2n+1]e^{j\frac{\pi(2n)}{N}}\right)e^{j\frac{2\pi(2n)k}{N}}\right]$$

$$= \sum_{n=0}^{N/2-1}\left[\left(x[2n+1]e^{j\frac{\pi(2n+1)}{N}}\right)e^{j\frac{2\pi(2n+1)k}{N}}\right]$$

Now it is straight forward to get G_k from Z_k.

$$\frac{1}{2}\left[Z_k + Z^*_{\frac{N}{2}-k-1}\right] + \frac{e^{j\frac{\pi(2k+1)}{N}}}{2j}\left[Z_k - Z^*_{\frac{N}{2}-k-1}\right]$$

$$= \sum_{n=0}^{N/2-1}\left[\left(x[2n]e^{j\frac{\pi(2n)}{N}}\right)e^{j\frac{2\pi(2n)k}{N}}\right] + \sum_{n=0}^{N/2-1}\left[\left(x[2n+1]e^{j\frac{\pi(2n+1)}{N}}\right)e^{j\frac{2\pi(2n+1)k}{N}}\right]$$

$$= \sum_{n=0}^{N-1}\left[\left(x[n] * e^{j\frac{\pi n}{N}}\right)e^{j\frac{2\pi nk}{N}}\right]$$

$$= G_k$$

Accordingly, use of an N/2 - Point FFT allows Gk to be derived, which inturn allows the MDCT of the complex input signal to be evaluated, with reduced computational requirements.

Claims:

1. A method of evaluating a modified discrete cosine transform (MDCT) of an input sequence of complex data using a Fast Fourier Transform (FFT), wherein the FFT is a N/2 - Point FFT.

2. A method as claimed in claim 1, wherein the MDCT is of the form:

$$X_k = \sum_{n=0}^{n=N-1} x[n] * \cos(2\pi * (2n+1) * (2k+1)/4N + \pi * (2k+1)/4) \quad k = 0...(N/2-1)$$

where $x[n]$ represents the input sequence and N is the number of data samples in the sequence.

3. A method as claimed in claim 2, wherein a transform coefficient sequence of the MDCT is derived using the N/2 - Point FFT of a complex function ($z[n]$, where

$$z[n] = (x[2n] + jx[2n+1])e^{j\frac{2\pi n}{N}}, \quad 0 \le n \le (N/2-1)$$

4. A method as claimed in claim 3, wherein a transform Zk and the complex conjugate of the transform $Z_{N/2-k-1}$ are derived from Z[n] for the expression

$$G_k = \frac{1}{2}\left[Z_k + Z^*_{\frac{N}{2}-k-1}\right] + \frac{e^{j\frac{\pi(2k+1)}{N}}}{2j}\left[Z_k - Z^*_{\frac{N}{2}-k-1}\right]$$

where Gk is a transform coefficient sequence which allows the MDCT to be evaluated, in that

$$G_k = g_{k,r} + jg_{k,i} = \sum_{n=0}^{n=N-1} (x[n]e^{j\pi n/N}) * e^{j2\pi nk/N}$$

which can be used to solve the MDCT when rewritten as

$$X_k = \cos\gamma(k)\ \underbrace{1/2(e^{j\pi(k+1/2)/N}G_k + e^{-j\pi(k+1/2)/N}G^*_k)}_{T_1} - \sin\gamma(k)\ \underbrace{1/2j(e^{j\pi(k+1/2)/N}G_k - e^{-j\pi(k+1/2)/N}G^*_k)}_{T_2}$$

$$= \cos\gamma * \underbrace{(g_{k,r}\cos(\pi(k+1/2)/N) - g_{k,i}\sin(\pi(k+1/2)/N)}_{T_1}$$

$$-\sin\gamma * \underbrace{(g_{k,r}\sin(\pi(k+1/2)/N) + g_{k,i}\cos(\pi(k+1/2)/N)}_{T_2}$$